AFRL-IF-RS-TR-2003-10
Final Technical Report
January 2003

TERABIT BURST SWITCHING

Washington University

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. F494/J162

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2003-10 has been reviewed and is approved for publication

APPROVED: 

ROBERT L. KAMINSKI
Project Engineer

FOR THE DIRECTOR: 

WARREN H. DEBANY, Technical Advisor
Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	JANUARY 2003	Final Jun 97 – Dec 02

4. TITLE AND SUBTITLE
TERABIT BURST SWITCHING

5. FUNDING NUMBERS
C - F30602-97-1-0273
PE - 62301E
PR - F494
TA - 00
WU - 01

6. AUTHOR(S)
Jonathan S. Turner

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Washington University
Department of Computer Science & Engineering
Campus Box 1045
1 Brookings Drive
St. Louis MO 63130-4899

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington Virginia 22203-1714

AFRL/IFG
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2003-10

11. SUPPLEMENTARY NOTES
AFRL Project Engineer: Robert L. Kaminski/IFG/(315) 330-1865/ Robert.Kaminski@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***
This is the final report for Washington University's Terabit Burst Switching Project, supported by DARPA and Rome Air Force Laboratory. The primary objective of the project has been to demonstrate the feasibility of Burst Switching, a new data communication service, which seeks to more effectively exploit the large bandwidths becoming available in WDM transmission systems. Burst switching systems dynamically assign data bursts to channels in optical data links using routing information carried in parallel control channels.

14. SUBJECT TERMS
Optical Switching, Optical Networking, Asynchronas Transfer Mode, ATM

15. NUMBER OF PAGES
32

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures / Tables

1. Project Summary

The primary objectives of the Terabit Burst Switching Project were to develop a new paradigm for optical switching called Optical Burst Switching (OBS), develop and evaluate OBS switch architectures and demonstrate OBS through the development and evaluation of a prototype system. A secondary objective was the development of a 160 Gb/s ATM switch, to be used in conjunction with the OBS demonstration system.

The project has made significant contributions to the development of optical burst switching concepts. Innovations developed for the project include the following:

- *Separation of burst control and data.* One of the key ideas behind the OBS concept is the separation of the burst control information from the burst header information. This allows burst headers to be processed electronically while data bursts pass through the switches transparently in the optical domain.
- *Variable burst offsets to compensate for control variation*. Our proposed OBS protocol incorporates a variable *offset* between each burst header cell and the burst that it controls. The use of a variable offset allows the offset value to be modified as bursts pass through the network to compensate for the inevitable delay variations present in the control subsystems of the burst switches.
- *Lookahead resource management*. In order for OBS networks to handle short bursts efficiently, it is necessary for the control subsystems of burst switches to construct schedules of projected resource usage and make control decisions based on these schedules.
- *Horizon scheduling.* We developed the first practical scheduling algorithm for OBS routers, showed how it could be implemented in hardware to provide very fast burst processing and determined conditions under which it provides optimal performance.
- *Horizon scheduling with reordering.* Horizon schedulers can perform poorly if there is large variability in the offsets between burst header cells and their corresponding bursts. This dependence of the performance on the offset can be eliminated by augmenting the horizon scheduler with a reordering buffer, allowing burst header cells to be processed in the order of burst arrival.

- *Differential search trees for burst storage management*. We developed an efficient data structure for maintaining a usage schedule for a burst storage subsystem. This data structure makes it possible to quickly determine if an incoming burst can be accommodated using the available memory and can be quickly updated to account for the arrival of new bursts.
- *Switch element design using tunable lasers and passive wavelength routers*. The cost of the datapath of optical burst switches is a key design issue. To reduce the cost of the datapath, we developed a novel switch element design whose only active component is a set of tunable lasers and showed how to analyze its blocking performance and demonstrated that it offers a practical alternative to more expensive nonblocking designs.
- *Time-sliced optical burst switching*. Wavelength conversion is a major cost component of optical burst switches. By switching bursts in the time domain instead of the wavelength domain, the costs can be reduced by a large factor. We are developing new architectures for such systems, that are arguably among the first designs for a form of optical packet switch that shows real potential for providing a cost-effective alternative to all-electronic routers.

In addition to the project's fundamental research activities, it sought to demonstrate optical burst switching and high capacity ATM switching through the development of two prototype systems. While the ATM switch was completed and has been successfully demonstrated, the prototype burst switch could not be completed, due to unexpected budget cuts that were made late in the program.

2. Personnel

A project of this magnitude requires a substantial engineering staff to turn the high level vision into working systems. We were fortunate to have a talented and dedicated engineering staff to carry out this complex effort. The table below lists the individuals who were involved and their roles in the project.

Name	Period	Role
Alex Chandra	2/99-2/02	hardware design engineer logic design for Time Stamp chip, ATM Interface Controller, Input Buffer chip and Output Buffer chip
Tom Chaney	10/97-6/02	Hardware Team Manager
Yuhua Chen	1/99-2/02	hardware design engineer Burst Processor logic design, Burst Switch Element board design, system architecture and performance evaluation
John DeHart	10/97-9/98, 1/02-3/02	software design engineer system planning, testing and evaluation
Maynard Engebretson	1/98-12/99	hardware design engineer physical design
J. Andrew Fingerhut	1/98-12/98	hardware design engineer Input Port Processor 2 specification

John Lockwood	6/99-9/00	hardware design engineer Burst Storage Unit, FPGA programming circuits
Tom McLaughlin	1/99-6/00	hardware design engineer Burst Processor logic design
Naji Naufel	2/99-1/00	hardware design engineer Switch Element logic design
Wenjing Tang	2/99-2/02	hardware design engineer physical design, printed circuit board layout for IO Module, ATM Interface, OC-48 line card
W. Dave Richard	3/98-3/01	hardware design engineer design of OC-48 line card, Dual G-link line card, Quad OC-12 line card
Michael Richards	1/98-1/00, 10/00-12/01	hardware technician printed circuit board layout for Burst Control board, Miscellaneous board, ATM Switch IO board, Dual G-link line card, Quad OC-12 line card
Fred Rosenberger	1/98-7/00	hardware design engineer Crossbar logic design, Sync. chip logic design
Randy Richards	3/98-10/01	hardware technician Input Port Processor logic design, printed circuit board layout for Switch Element board, Crossbar board, Burst backplane, ATM backplane

3. Burst Switching Concept

By some estimates, bandwidth usage in the Internet is doubling every six to twelve months [CO98]. Data network capacities now surpass voice network capacities and the growing demand for network bandwidth is expected to continue well into the next century. Current networks use only a small fraction of the available bandwidth of fiber optic transmission links. The emergence of WDM technology is now unlocking more of the available bandwidth, leading to lower costs, which can be expected to further fuel the demand for bandwidth.

We now face the near-term prospect of single fibers capable of carrying terabits per second of data. This leads to a serious mismatch with current switching technologies, which are capable of switching at rates of "only" 1-10 Gb/s. While emerging ATM switches and IP routers can be used to switch data using the individual channels within a WDM link (the channels typically operate at 2.4 Gb/s or 10 Gb/s), this approach implies that tens or hundreds of switch interfaces must be used to terminate a single link with a large number of channels. Moreover, there can be a significant loss of statistical multiplexing efficiency when the parallel channels are used simply as a collection of independent links, rather than as a shared resource.

Proponents of optical switching have long advocated new approaches to switching using optical technology in place of electronics in switching systems [BA97,GA97,MA96]. Unfortunately, the limitations of optical component technology [GU96,IK96,ST96] have largely limited optical switching to facility management applications. While there have been attempts to demonstrate the use of optical switching in directly handling end-to-end user data channels,

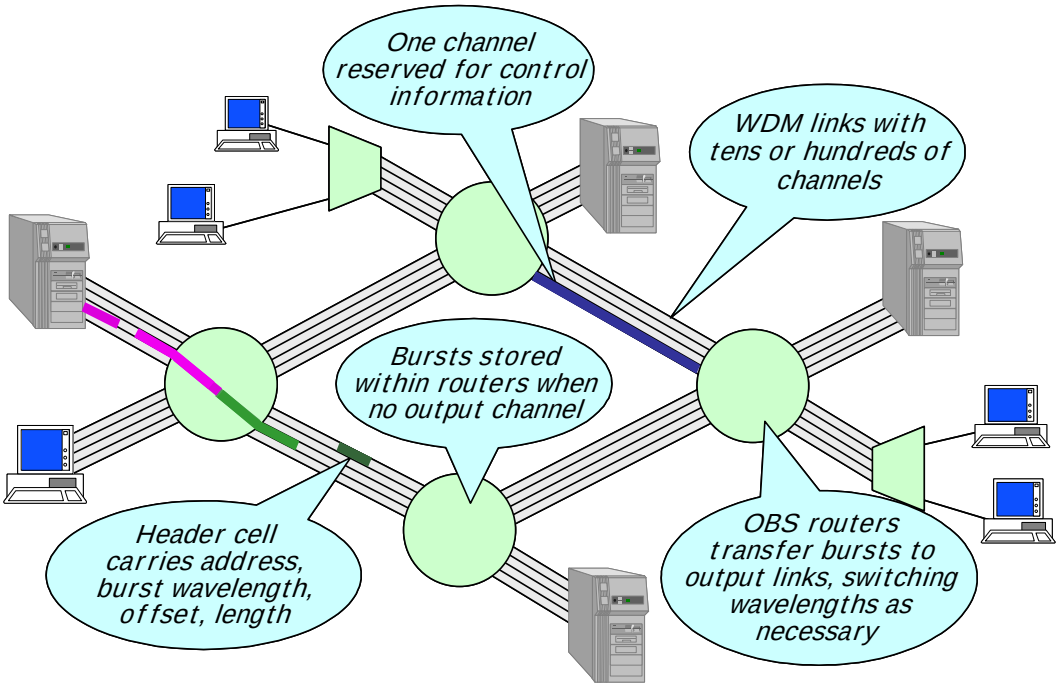

Figure 1. OBS Burst Switching Concept

these experiments have been disappointing. Indeed they primarily serve to show how crude optical components remain and have done little to stimulate any serious move toward optical switching.

This project has sought to develop an approach to high performance networking that can more effectively exploit the capabilities of fiber optic transmission systems and facilitate a transition to switching systems in which optical technology plays a more central role. *Optical Burst Switching* (OBS) is designed to make best use of optical and electronic technologies. It uses electronics to provide dynamic control of system resources, assigning individual user data bursts to channels on WDM links. The control mechanisms are designed to efficiently handle data bursts as short as a kilobyte or as long as many megabytes. OBS is designed to facilitate switching of the user data channels entirely in the optical domain. While current optical components remain too crude for this to be practical, anticipated improvements in integrated optics could ultimately make optical-domain switching feasible and economically viable.

Figure 1 shows the basic concept for an OBS network. The transmission links in the system carry multiple WDM channels, any one of which can be dynamically assigned to a user data burst. One (or possibly more than one) channel on each link is designated a *control channel*, and used to control dynamic assignment of the remaining channels to user data bursts. When an end system has a burst of data to send, an idle channel on the access link is selected, and the data burst is sent on that idle channel. Shortly before the burst transmission begins, a *Burst Header Cell* (BHC) is sent on the control channel, specifying the channel on which the burst is being transmitted and the destination of the burst. An OBS router, on receiving a BHC, selects an outgoing link leading toward the desired destination with an idle channel available, and then establishes a path between the specified channel on the access link and the channel selected to carry the burst. It also forwards the BHC on the control channel of the selected link, after modifying the cell to specify the channel on which the burst is being forwarded. This

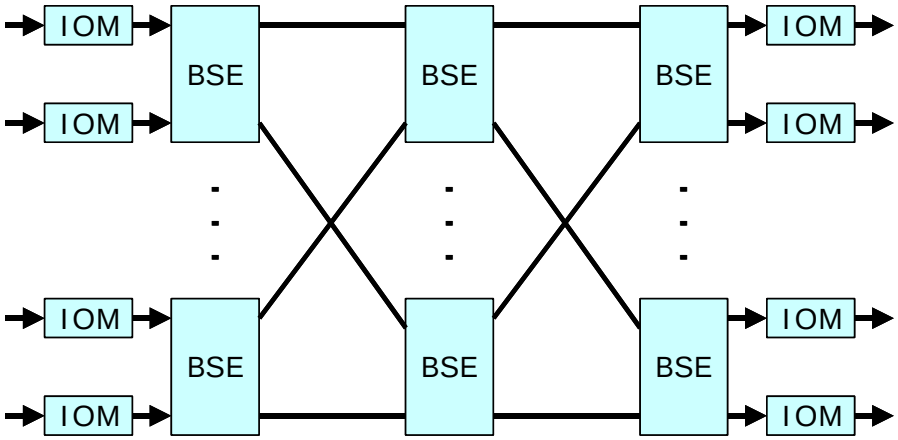

Figure 2. Scalable Burst Switch Architecture

process is repeated at every switch along the path to the destination. The BHC also includes a length field specifying the amount of data in the burst. This is used to release the path at the end of the burst. If, when a burst arrives, there is no channel available to accommodate the burst, the burst may be stored in a buffer or discarded if there is no storage space available. OBS routers include a control section, which processes BHCs and establishes paths for data bursts and a data path, which propagates the bursts in the optical domain. The separation of the control channels from the data makes it straightforward to keep the data path in the optical domain, while processing the BHCs electronically.

To handle short data bursts efficiently, OBS routers must maintain tight control over the timing relationships between BHCs and their corresponding data bursts. Uncertainty in the precise timing of the beginning and ending of bursts leads to inefficiencies, since OBS routers must allow for these uncertainties when setting up and tearing down paths. For efficient operation, timing uncertainties should be no more than about 10% of the average burst duration. For efficient handling of bursts with average lengths of 1 KB or less on 10 Gb/s channels, the end-to-end timing uncertainty in a burst network must be limited to less than 100 ns. To enable precise timing of switching operations, each BHC includes an *Offset* field, which specifies the time between the transmission of the first bit of the BHC and the transmission of the first bit of the data burst. Although OBS routers typically delay data bursts by a constant amount of time (in the common case where bursts do not need to be buffered), BHCs may experience variable delays due to contention and queueing within the electronic control subsystem that processes the BHCs. To account for this, BHCs are timestamped when they enter an OBS router and the timestamps are used to determine the amount of time they have been delayed when they are forwarded on the outgoing link. This allows the outgoing offset value to be updated to reflect the variable delay. This mechanism also makes it straightforward to adjust for the variable delays that bursts experience when transmitted over channels with different wavelength-dependent delays.

4. Scalable OBS Router Architecture

Electronic routers with terabit capacities are now becoming technically feasible and commercially available. While optical switching may provide an alternative to electronics in this performance range, it will provide the greatest advantage in systems that are beyond the reach

of electronic switching. For this reason, it is important to consider scalable architectures capable of providing petabit capacities. Figure 2 shows a scalable OBS router architecture consisting of a set of *Input/Output Modules* (IOM) that interface to external links and a multistage interconnection network of *Burst Switch Elements* (BSE). The interconnection network uses a Beneš topology, which provides multiple parallel paths between any input and output port. A three stage configuration comprising d port switch elements can support up to d^2 external links (each carrying many WDM channels). The topology can be extended to 5, 7 or more stages. In general, a $2k-1$ stage configuration can support up to d^k ports, so for example, a 5 stage network constructed from 8 port BSEs supports 512 ports. If each external link carries 256 WDM channels at 10 Gb/s each, the aggregate system capacity exceeds one petabit per second.

The control section of each IOM terminates the control channel (or channels), converting it to electronic form, so that it can process the BHCs. The IOM uses the address information in the BHCs to do a routing table lookup. The result of this lookup includes the number of the output link that the burst is to be forwarded to. This information is inserted into the BHC, which is then forwarded to the first stage BSE. The data channels pass transparently through the IOMs but are delayed at the input using a fixed length fiber delay line. This delay allows time for the control operations performed in the IOM and within the interconnection network that follows.

When a BHC is passed to a BSE, the control section of the BSE uses the output port number in the BHC to determine which of its output links to use when forwarding the burst. If the required output link has an idle channel available, the burst is switched directly through to that output link. If no channel is available, the burst can be stored within a shared *Burst Storage Unit* (BSU) within the BSE. Burst storage can be provided in all stages of a burst network, or can be limited to the output stage. In this latter case, it can be beneficial to provide extra bandwidth capacity on the inter-stage links joining BSEs together. This can be accomplished using extra wavelengths or by providing extra fibers. In the latter case, the network topology becomes that of a Clos network.

In the first $k-1$ stages of a $2k-1$ stage network, bursts can be routed to any one of a BSE's output ports. The port selection is done dynamically on a burst-by-burst basis to balance the traffic load throughout the interconnection network. This use of *dynamic routing* yields optimal scaling characteristics, making it possible to build large systems in which the cost per port does not increase rapidly with the number of ports in the system.

At the output IOM, the BHC is forwarded on the outgoing link and the offset field is adjusted to equal the time delay between the transmission of the first bit of the BHC and the first bit of the burst. The outgoing IOM may delay BHCs in order to reduce the variability in the offset values seen by downstream routers.

There are several approaches that can be used to implement multicast switching in an architecture of this type. However, the most efficient approach implements multicast in multiple passes through the network, with binary copying in each pass. To implement this method, a subset of the system's inputs and outputs are connected together in a loopback configuration. Ports connected in such a loopback configuration are called recycling ports.

The routing table in an IOM can specify a pair of output ports that a burst is to be forwarded to, and the multistage interconnection network uses this information to forward the burst to both of the specified outputs. If either or both of the specified outputs is a recycling port, the corresponding copy of the burst will pass through the system again, allowing it to be forwarded

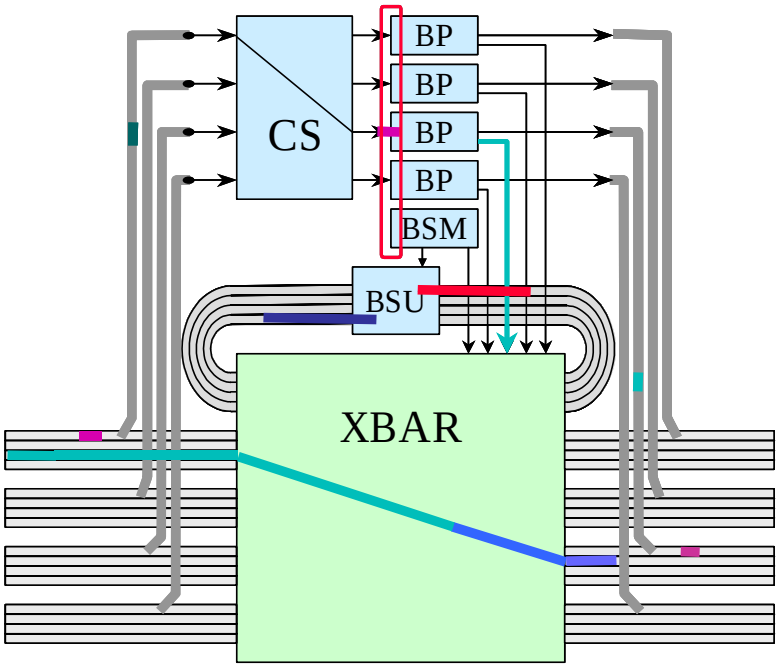

Figure 3. Burst Switch Element

to two more outputs. In this way, a burst can be forwarded to four outputs in two passes through the network, to eight outputs in three passes, etc.

The total bandwidth consumed by multicast traffic on the recycling ports is strictly less than the total bandwidth consumed on output links that are not recycling ports. If δ is the fraction of the outgoing traffic that is multicast, then a system with m output links will need δm recycling ports. Since δ is typically fairly small (say .1), the incremental cost of multicast is also small. In any case, the incremental cost of multicast switching is just a (small) constant factor over the cost of a comparable system that only supports unicast. All known single pass multicast methods, on the other hand, lead to a multiplicative cost increase that grows at least as fast as the logarithm of the number of ports. The multipass approach also makes it easy to add endpoints to, or remove endpoints from a multicast session, and can be used to provide scalable many-to-many multicast, as well as one-to-many. This method has been adapted from the ATM switch architecture described in [CH97]; further details can be found there and in references [TU94, TU96].

The Burst Switch Element is the key component of the OBS router. Figure 3 shows how the BSE can be implemented. In this design, the control section consists of a d port Cell Switch (CS), a set of d *Burst Processors* (BP), and a *Burst Storage Manager* (BSM). The data path consists of an optical wavelength converting *Crossbar* (XBAR), together with a *Burst Storage Unit* (BSU). The BSU is connected to the crossbar with m input links and m output links. The crossbar is capable of switching a signal on any channel within any of its input links to any channel within any of its output links; so in a system with d input and output links and h data channels per link, we require the equivalent of a $(d+m)h \times (d+m)h$ crossbar. The BSU can be simply a collection of fiber delay lines of varying length or a more complex optical storage subsystem. Alternatively, electronic storage can be used, although in this case it's necessary to sacrifice the objective of optical transparency.

Each BP is responsible for handling bursts addressed to a particular output link. When a BP is unable to switch an arriving burst to a channel within its output link, it requests use of one of the BSU's storage locations from the BSM, which switches the arriving burst to an available storage location (if there is one). Communication between the BSEs and the BSM occurs through a local *control ring* provided for this purpose.

5. Control of Burst Switch Elements

The design of the control subsystem of the Burst Switch Elements is one of the key issues for OBS routers. The control mechanisms implemented by the Burst Processors must support efficient use of the outgoing links while being simple enough to allow for very high speed processing. A system with 256 channels per link, each operating at 10 Gb/s, must process over 300 million BHCs per second, if it is to accommodate average burst lengths of 1 KB. Even if the average burst length is relaxed to 10 KB, the required processing rate remains very demanding. Achieving such rates requires scheduling mechanisms that are simple, have low computational complexity and can make effective use of hardware parallelism.

To provide efficient handling of short bursts, it's important for BPs to allocate link bandwidth to a burst only during the time period that the burst is actually passing over the link. It cannot simply allocate a channel to a burst at the time the BHC is received, since BHCs can arrive well in advance of their bursts. This leads to the concept of *Lookahead Resource Management* in which link (and storage) resources are scheduled in advance, using knowledge of the projected channel usage at the time bursts are expected to arrive.

5.1. Horizon Scheduling

Horizon scheduling is a particularly straightforward approach to managing resource use in an OBS router. A BP that uses horizon scheduling, maintains a single *scheduling horizon* for each channel on its outgoing link. The scheduling horizon for a channel is defined as the latest time at which the channel is currently scheduled to be in use. Given this information, the procedure for assigning a burst to a channel becomes obvious. Simply select from among the channels whose scheduling horizons precede the burst's arrival time and select the channel from this set with the latest scheduling horizon. Once a channel has been selected, recompute the scheduling horizon to be equal to the time when the burst is due to be completed (determined using the offset and length information in the BHC). If no channels have horizons that precede the arrival time of the burst, then the burst can be discarded or can be stored, with the assistance of the BSM. We defer discussion of storage management until section 5.5.

The great advantage of horizon scheduling is that it can be implemented to operate at very high speed, using hardware parallelism. For each channel, we need a register storing a channel number and the scheduling horizon for that channel, plus a comparator. The (channel number, horizon) pairs are ordered by the horizon values. This makes it easy to find the channel with the largest horizon preceding the burst arrival time (all horizons are compared to the burst arrival time in parallel, and the last one in the list that has a smaller horizon is selected). Using modern ASICs, the required operations can be completed in under 10 ns. What makes this fast hardware implementation possible is the fact the amount of circuitry required grows in proportion to the number of channels (not the number of bursts). This makes it feasible to implement using only highly parallel, on-chip logic, since the number of channels is inherently limited. Scheduling methods whose complexity is proportional to the number of bursts typically must use off-chip

memory to store the required per-burst state information. Retrieving this information quickly when it is needed can be difficult to do.

5.2. Horizon Scheduling with Reordering

Horizon scheduling can be very efficient if the variance in the offsets of arriving bursts is small. In particular, if the difference between the largest and smallest offsets is smaller than the shortest burst duration, then any set of bursts that can be scheduled without storage can be scheduled by a horizon scheduler (without requiring storage). Unfortunately, considerable offset variation can occur in an OBS network. Each OBS router includes a fixed delay in its datapath, to allow for the variable delays that can occur in the control subsystem. The datapath delay is chosen so that with very high probability, the BHC processing can be done within the time allowed by the fixed delay. This means that in lightly loaded networks, offsets can increase significantly as bursts are propagated through multiple routers. While BHCs can be artificially delayed at the output ports of routers to limit the growth in the offsets, significant variation in offset values can still be expected.

Because horizon scheduling does not keep track of the idle periods between scheduled bursts, it may not be able to schedule some bursts that could be scheduled during those idle periods. In particular, when a BP receives a BHC with an unusually large offset, its horizon scheduler must select an outgoing channel for the burst, leaving a large gap between the current time and the start time of the burst. If the BP later receives a BHC for a burst whose time span falls within that gap, the horizon scheduler will be unable to schedule it using the same channel, possibly forcing it to discard the burst, unnecessarily.

The addition of *reordering* to a horizon scheduler can yield significantly better performance. In a horizon scheduler with reordering, BHCs are processed in the order of burst arrival, not in the order of BHC arrival. This is implemented by passing BHCs through a resequencing buffer and holding them there, until some fixed time period before the expected burst arrival time (this time period is called the *deadline*). Because the number of BHCs that must be held in the resequencing buffer is potentially much larger than the number of channels, the use of off-chip memory may be needed to store the waiting BHCs. Fortunately, the nature of the resequencing operation is simple enough that the use of off-chip memory does not create a serious performance bottleneck in this case. The data structure used to implement resequencing requires just a constant number of memory accesses to enqueue an arriving cell and retrieve the next cell to go out. With a 16 byte BHC payload, a single 32 bit wide DDR SRAM with a clock rate of 133 MHz provides sufficient bandwidth to resequence more than 25 million BHCs per second. Using four such memories in parallel, the processing rate can be improved to over 100 million per second. The performance can be expected to scale directly with continuing improvements in memory bandwidth.

The performance of horizon scheduling with reordering is a function of the deadline used to trigger the scheduling operation. If the deadline is no larger than the sum of the minimum offset and the minimum burst duration, then a reordering scheduler can schedule, without requiring buffering, any set of bursts that can be scheduled without buffering (by any scheduling algorithm). Unfortunately, delaying the processing of BHCs until the deadline is reached can also have a negative effect on the throughput of the BP. If BHC processing is delayed too long, some bursts may have to be discarded because the BHCs do not get processed until after the burst has arrived. Consequently, the deadline has to be chosen with this trade-off in mind. Alternatively, adaptive deadline adjustment can be used to automatically adjust the deadline so

that it's small enough to ensure that most BHCs are processed in order of burst arrival, and large enough to ensure that BHCs rarely fail to get processed by the time a burst arrives.

5.3. Storage Management

Optical burst switches supporting links with large numbers of wavelengths can operate at high levels of utilization with only a very small probability that arriving bursts must be discarded. In systems where the number of wavelengths is more limited, buffering can yield significantly better performance. This can be particularly useful in access networks where the traffic may not be sufficient to support large numbers of wavelengths. To handle such situations, the BSE design described earlier includes a Burst Storage Unit (BSU) to provide temporary storage for bursts that cannot be sent directly to an output channel and a Burst Storage Manager (BSM) to control the use of the BSU.

The storage management mechanism implemented by the BSM must be tailored to the nature of the storage provided by the BSU. In the simplest case, the BSU consists of a set of delay lines of various lengths. In this case, the scheduling of a burst requiring storage consists of the following steps.

- The BP at the outgoing link for the burst considers its set of outgoing channels. If all of the channels have horizons that are later than the arrival time of the burst, the BP determines earliest horizon (call this value *h*) among its channels and issues a request to the BSM. The storage request specifies the arrival time of the burst and requests that it be sent back to the BP as soon as possible after time *h*.
- The BSM uses horizon schedulers to schedule each of its delay lines. It selects the delay line with the smallest delay that is long enough to provide the required delay. If this delay line has no available channels it considers delay lines with larger delays until it finds one that has an available channel. The BSM then issues a reply to the BP, informing it of the delay that it will provide (or indicates that it cannot accommodate the burst if none of its delay lines has an available channel).
- On receiving the reply, the BP selects the channel with the earliest horizon preceding the time when the burst will return from the BSM and assigns the burst to this channel.

This algorithm is straightforward to implement at high speed and is directly compatible with the most practical form of optical storage. The performance and cost are determined by the total length of the delay lines, the number of delay lines and the choices of delay values provided. If the total delay line length and the number of delay lines are fixed, we have found that superior performance is obtained when delay lines lengths form a geometric progression rather than a linear progression. The geometric distribution of lengths allows a larger range of delay values to be provided with a given total delay line length. With a geometric distribution, the largest delay value grows in proportion to the total delay line length, while a linear distribution of delay line lengths yields a largest delay value that grows in proportion to the square root of the total delay line length.

In the above description, it is assumed that the optical crossbar that switches bursts among inputs, outputs and the BSU fibers provides wavelength conversion for the BSU fibers, just as it does for the I/O channels. This is not strictly necessary. Omitting wavelength conversion from the BSU fibers has the potential to yield a significant cost savings, but can have a negative

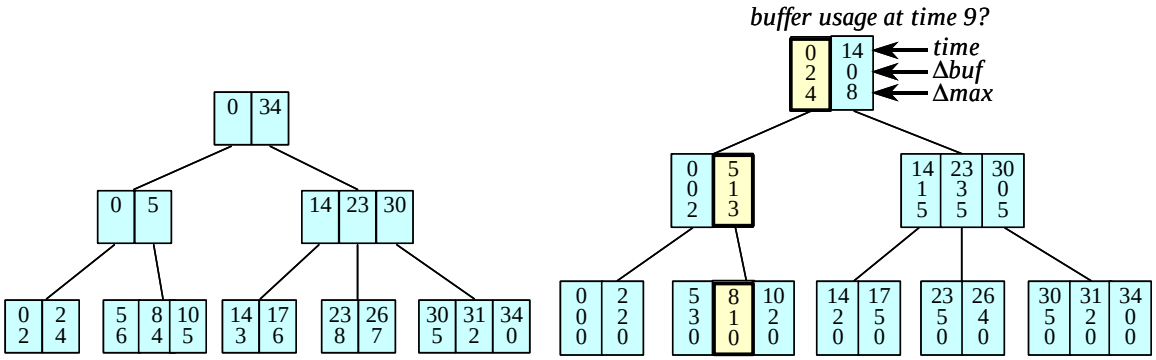

Figure 4. Basic Search Tree (left) and Differential Search Tree (right).

impact on the performance of the BSU. To compensate, it may be necessary to increase the number of delay lines and / or their total length.

While optical delay lines provide a particularly simple form of burst storage, they provide fairly limited flexibility. More sophisticated optical storage subsystems can store bursts for arbitrary amounts of time (rather than for fixed time intervals) and are less subject to contention at the interface entering the BSU. In systems where optical transparency is not required, electronic storage can be a viable alternative, offering fully flexible storage and relatively low cost per bit (less than 10% the cost of optical delay line storage). While electronic storage also requires electro-optic conversion, the associated cost is no greater than the cost of wavelength conversion.

A more flexible storage subsystem allows the provided memory to be used more efficiently. To manage the storage, we require a data structure that represents the projected usage of the memory. We define the *buffer usage function* to be a function $b(t)$ that represents the amount of buffer space that is scheduled to be in use at time t. The buffer usage function is a piecewise linear function that changes value, stepwise, at times corresponding to the arrival and departure of bursts and remains constant at other times. It can be represented by a set of pairs (t,b) where t corresponds to a time where the function value changes and b is the value of $b(t)$ immediately after the change at time t. If these pairs are stored in a balanced search tree, we can find the buffer usage value at any time t in $O(\log n)$ time, where n is the number of bursts that are stored in the BSU. An example of such a data structure (based on a 2-3 tree) is shown on the left side of Figure 4. Each leaf contains two or three pairs (t,b), and the leaves are ordered by their time values. The internal nodes of the data structure contain two or three time values equal to the smallest time value within one of the node's subtrees. This enables rapid searches for the desired time interval. For scheduling the use of the BSU, we need more than the ability to determine the buffer usage at a particular time. In particular, we need to be able to quickly determine the *maximum* buffer usage in a time interval $[t_1,t_2]$. We also need the ability to change the value of the buffer usage curve by a fixed amount over a specified time interval. These capabilities can be provided using the *differential search tree*, illustrated on the right side of Figure 4. Each node of the differential search tree contains the same time value as the basic search tree. It also contains two additional fields Δbuf and Δmax. The Δbuf values are chosen so that the sum of the Δbuf values on the path from the root to a leaf entry gives the value of $b(t)$ during the time interval represented by that leaf entry. Thus, we can still quickly determine a value of $b(t)$ by summing the Δbuf values as we search down from the root. Moreover, we can effectively change the $b(t)$ values for a subtree by a constant amount, simply by changing the Δbuf value for that subtree. This makes it possible to modify the buffer usage over an arbitrary

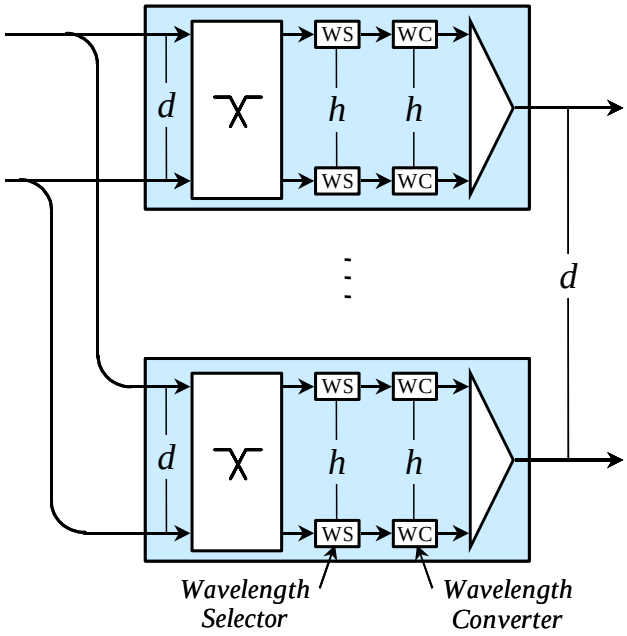

Figure 5. Space / Wavelength Switch

time interval in $O(\log n)$ time. The Δmax values are chosen so that by adding the Δmax value for a particular subtree to the sum of the Δbuf values from that entry to the root of the tree, we get the maximum value of $b(t)$ during the time interval represented by the subtree. This makes it possible to determine the maximum value of $b(t)$ during an arbitrary time interval in $O(\log n)$ time.

6. Data Path Architectures for Optical Burst Switching

While not part of the original scope of this project, we have also studied two major alternatives for implementing the optical data path of a burst switch element, in order to get a better understanding of the design trade-offs and cost implications for optical burst switching. The first of the two architectures considered is a space / wavelength architecture, in which switching is performed first in the space dimension and then in the wavelength dimension. The second approach is a wavelength / space architecture, in which switching is performed first in the wavelength dimension. For this architecture, we have explored two variants, one of which uses passive wavelength routers in place of active switching components.

6.1. Space/Wavelength Architecture

The crossbar at the center of a Burst Switch Element is a wavelength converting switch that allows a signal carried on any input fiber and input wavelength to be forwarded to any output fiber and output wavelength. This can be implemented using a space / wavelength architecture in which switching is done first in the space dimension, then in the wavelength dimension (Figure 5). Each of the d input fibers is first passively split, so that a copy of all the input signals is sent to each of a set of d output sections. Each of the output sections selects a subset of the input signals, converts the selected signals to a compatible set of wavelengths and multiplexes them onto its output link. Each output section contains a d input, h output optical crossbar (h is the number of wavelengths on each fiber), followed by a set of *wavelength selectors* and

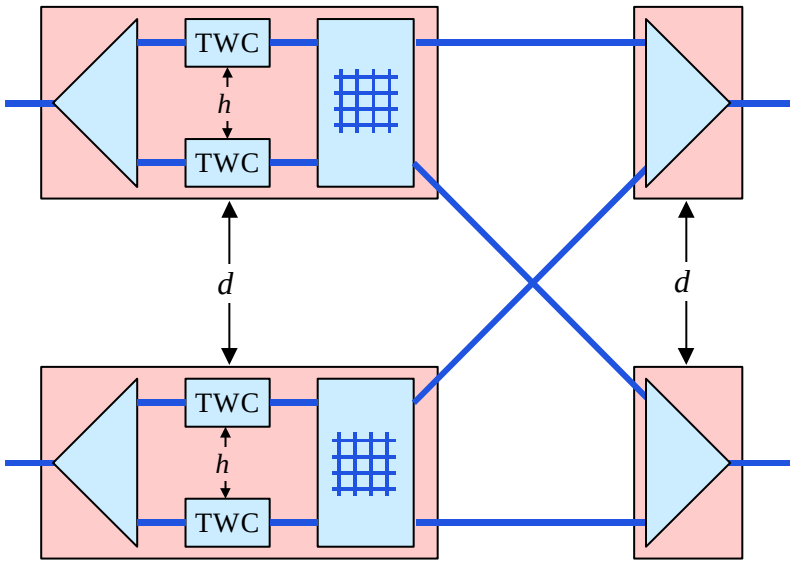

Figure 6. Wavelength Switch Using Tunable Wavelength Converters (TWC), Optical Crossbars and Passive Multipexors and Demultiplexors

wavelength converters. The wavelength selectors can be viewed as tunable filters. They allow a specified wavelength to be propagated to the output, while all others are suppressed. Each wavelength converter changes its received input signal to a fixed output wavelength.

The dominant cost components of the space / wavelength architecture appear to be the wavelength selectors and converters. A straightforward implementation of a wavelength selector consists of an optical demultiplexor, followed by an optical gate for each of the demultiplexed channels (these can be implemented using semiconductor optical amplifiers), followed by a passive optical coupler to propagate the selected signal to the output. The largest contributor to the cost is the *h* optical gates needed for each outgoing channel. The wavelength converter can be implemented using a Mach-Zender interferometer (or similar all-optical conversion method) to allow the input signal to modulate an optical carrier on a fixed output wavelength. The main cost component here is the laser needed to provide the fixed carrier. This cost may be reduced by sharing lasers of the same wavelength among the different output sections of the BSE.

6.2. Wavelength/Space Architecture

An alternative to the space / wavelength architecture is a wavelength / space architecture, in which switching is performed first in the wavelength domain, using wavelength converters that can be tuned to a specified output wavelength. One such architecture is shown in Figure 6. Each of the *d* input sections contains an optical demultiplexor, followed by a set of *h Tunable Wavelength Converters* (TWC), followed by a crossbar with *h* inputs and *d* outputs, where *h* is the number of wavelengths per link and *d* is the number of inputs and outputs of the burst switch element. Typically *h* will be fairly large (expected values would range from 64 to 256), while *d* will be relatively small (8 or 16 possibly). Each *h*×*d* crossbar can be decomposed into a set of *d*×*d* crossbars, followed by a set of passive multiplexors, so systems of practical interest can be built using a crossbar technology capable of producing 8×8 crossbars, for example. One thing to note

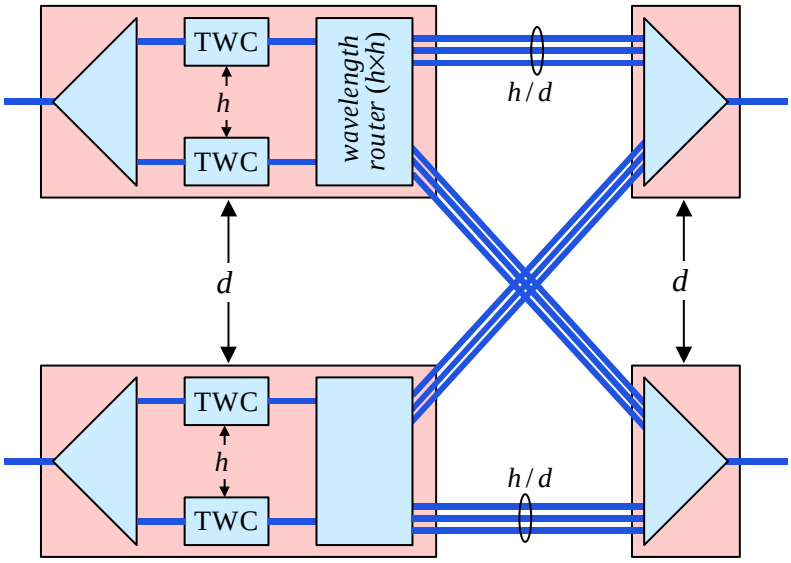

Figure 7. Wavelength Switch Using Tunable Wavelength Converters (TWC) and Passive Wavelength Routers (AWGN)

about this design is that the crossbars really implement both a space division switching function and a multiplexing function. If several of the input channels on a given input link are routed to the same output link, they will be multiplexed onto the connecting link between the input and output sections. SOA-based crossbar technologies can implement this combined switching and multiplexing function. The output sections in this design are simply passive multiplexors. The two dominant cost components are the optical switching components and the tunable wavelength converters, each of which comprises a tunable laser and an optical modulator. Recent progress on tunable lasers may allow these devices to be comparable in cost to fixed wavelength lasers.

An alternative design is shown in Figure 7. This design uses a passive wavelength router (AWGN-type) in place of the optical crossbars used in the first switch design. Thus, the tunable wavelength converters are the only active components. Since the wavelength routers have h inputs and h outputs, h/d fibers connect each input section with each output section (alternatively, these signals can be passively coupled to a single outgoing fiber). For h=256 and d=8, there will be 32 fibers connecting each input section with each output section. In this design, the tunable wavelength converters serve two purposes. First, they provide the required space switching. By tuning the laser to one wavelength in the appropriate set of h/d wavelengths, we can "steer" the signal to the desired output port. At the same time, we need to avoid wavelength conflicts on the output links of the system, so the choice of output wavelengths is constrained. The implication of this, is that this design, is not nonblocking. That is, there may be situations where all of the wavelengths that can be used to get to a desired output, are in use, causing blocking to occur, even when there are free wavelengths available on the outgoing link.

The likelihood of blocking in these systems is largely determined by the pattern of interconnections used to connect the input sections with the output sections. If we select this pattern appropriately, we can dramatically increase the number of wavelengths that will be

available to route from a set of inputs to any given output. While any individual input channel has h/d wavelengths it can use to reach an output, a pair of input channels may have close to $2h/d$ wavelengths that they can use to reach a given output. We can reduce the likelihood of blocking by using an interconnection pattern for which different input channels share only a few wavelengths for reaching any given output. We have developed a method of analyzing such interconnection patterns and have shown that for systems of practical interest, it is possible to achieve performance levels that are roughly comparable to those achieved with fully nonblocking designs.

6.3. Fundamental Issues with Wavelength Switching Architectures

Over the course of this project, it has become clear that there are some fundamental issues with OBS architectures that require wavelength conversion. All known techniques for wavelength conversion require the equivalent of a laser and an optical modulator. This is required for each outgoing channel. In a large multistage router, the conversion is required at every stage. This makes it difficult for OBS routers to be cost-competitive with electronic routers. The reason for this is that the largest single contributor to the cost of an electronic router for WAN applications is the set of components that implement the required electo-optic conversions. These components can account for as much as half the parts cost of a WAN router and these costs are dominated by the cost of the laser and modulator. This makes it impossible for an OBS router to have a significant cost advantage, relative to an electronic router, and since OBS routers inherently provide less functionality than electronic routers, it's hard to see how one might justify their commercial deployment. For optics to gain a decisive edge, the cost of the electro-conversion must be dramatically reduced, to where it becomes just a small fraction of the cost of the electronic router. This requires improvements in the cost-performance of optical components at a rate that is faster than the continuing improvements in electronics. While such improvements are not out of the question, there seems little reason to expect them.

7. Time Sliced Optical Burst Switching

To address the growing concerns with the high cost of wavelength conversion, we have developed a new variant of optical burst switching called *Time-Sliced Optical Burst Switching* (TSOBS), which replaces switching in the wavelength domain with switching in the time domain. By using a time-slotted link format TSOBS makes it possible to do time-domain switching without large amounts of optical buffering. We have found that an appropriately designed TSOBS router requires less than 1% of the storage that would be required using a more conventional packet switching architecture.

As in ordinary burst switching, TSOBS separates burst control information from burst data. Specifically, *Burst Header Cells* (BHC) are transmitted on separate control wavelengths on each WDM link. These wavelengths are converted to electronic form at each switch, while all remaining wavelengths are switched through in optical form. The data wavelengths carry information in a Time-Division Multiplexed (TDM) format, consisting of a repeating *frame structure*, which is sub-divided into *time slots* of constant length. A repeating sequence of time slots in successive frames, at a fixed position within the frame constitutes a *channel* that can be used to carry an end-to-end data burst. Each BHC "announces" the imminent arrival of a data burst, and includes address information plus the wavelength and channel on which the burst is arriving. It also includes an *offset*, which identifies the frame in which the first timeslot

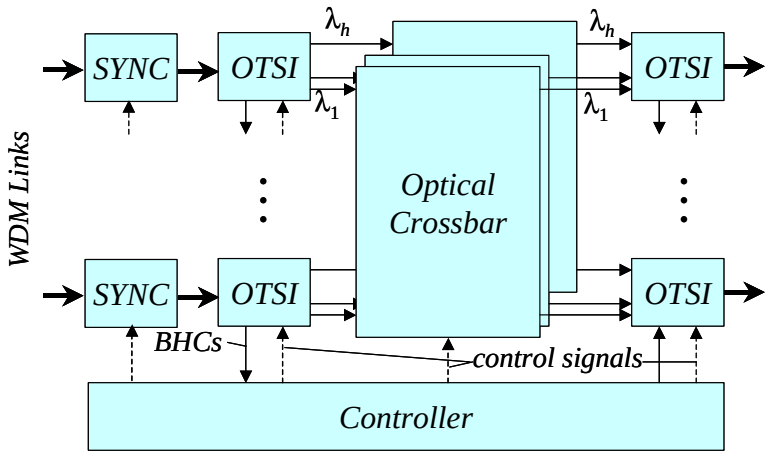

Figure 8. TSOBS Router

containing data from the burst appears, and a *length*, which identifies the number of timeslots used to transmit the burst.

Optical Time Slot Interchangers (OTSI) are the key building blocks of routers in TSOBS networks. Three key factors that affect the cost and performance of an OTSI are (1) the size of its internal crossbar, (2) the amount of fiber required for the delay lines used to reorder the timeslots, and (3) the number of switching operations that bursts may be subjected to when passing through the OTSI. We have developed an overall architecture for a TSOBS router and have studied how alternative OTSI designs affect its cost and performance.

Figure 8 shows the overall design for a TSOBS router. Each incoming WDM link terminates on a *Synchronizer* (SYNC) which synchronizes the incoming frame boundaries to the local timing reference. This is done using variable delay lines, with feedback control of the delays being provided through the system controller. The synchronizers are followed by Optical Time Slot Interchangers (OTSI), which provide the required time domain switching for all wavelengths. The OTSIs also separate the control wavelengths carrying the BHCs and forward those to the system controller.

In addition the input OTSIs separate the data wavelengths and forward these on separate fibers to each of a set of Optical Crossbars at the center of the diagram. The crossbars perform the required space division switching operation. A second set of OTSIs is provided at the output. These provide another stage of time domain switching and remultiplex the data wavelengths and control wavelengths (carrying the outgoing BHCs) on the output fibers. The controller uses the information in the BHCs to make switching decisions and generates electronic control signals, which are used to control the operation of the OTSIs and the crossbars.

Figure 9 shows a high level design for one of the OTSIs. Each OTSI contains a set of optical crossbars for switching timeslots among the inputs, outputs and a set of delay lines. The signals are demultiplexed to perform the switching operations and re-multiplexed onto the delay lines, allowing the cost of the delay lines to be shared by the different wavelengths. The number of delay lines and the choice of delay line values are key design parameters, significantly affecting both the cost and performance of the OTSI.

We can classify OTSI designs as either blocking or nonblocking. While nonblocking designs provide the best performance, they are significantly more expensive than blocking designs. We

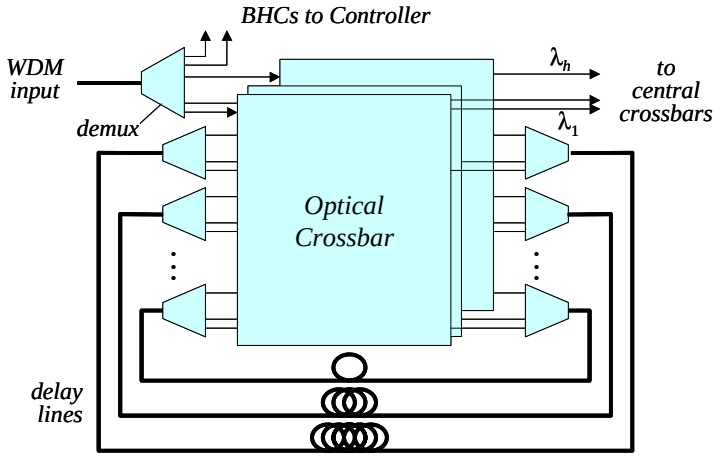

Figure 9. Optical Time Slot Interchanger

start with the conceptually simplest nonblocking design, which has N delay lines with a delay value equal to the duration of one time slot. With this design, each incoming timeslot i can be delayed by d timeslot intervals by recirculating it through the i-th delay line d times. Since each timeslot is assigned to a separate delay line, there are no conflicts, hence the design is nonblocking. It also uses the smallest possible total delay line length (N, where the unit is the distance light propagates in one timeslot interval). Unfortunately, it requires a large number of separate delay lines (N) and large optical crossbars ((N+1) × (N+1)). The optical crossbars are a particular concern since their cost grows as the product of the number of inputs and outputs.

Finally, the design can subject a signal to up to N optical switching operations, causing excessive degradation to the optical signal quality, when N is large. This last fault can be corrected by replacing the delay lines of length 1, with delay lines of length 1,2, . . . ,N. This allows each timeslot to be switched through just a single delay line, reducing the number of switching operations to 2. Of course, it comes at the cost of increasing the total delay line length from N to approximately $N^2/2$.

A more practical nonblocking OTSI design uses delay lines of length 1, 2, 3, . . . , (A–1), where A is an integer parameter, plus additional delay lines of length A, 2A, 3A, . . . , (B–1)A time slots, where B is a second integer parameter. We call these two sets of delay lines the *short delay lines* and the *long delay lines*. Let us suppose a time slot has to be delayed by a value of D time slots. D can be expressed as a sum, $k_2A + k_1$, where k_1 lies in the interval [0,A) and k_2 lies in the interval [0,B). To delay the time slot by D, we pass the data through the long delay line of length k_2A and then pass it through the short delay line of length k_1. The maximum we can delay a signal using this configuration is (B–1)A+(A–1) and since the maximum delay needed is N–1, this gives us the relation $AB \geq N$. The number of delay lines in this design is A+B–2 and hence, choosing $A = B = \lceil N^{1/2} \rceil$ gives us the minimum number of delay lines. This design can be used to delay a time slot for any time interval in the range and yields a nonblocking OTSI.

Blocking OTSIs are an alternative to nonblocking OTSIs, offering lower complexity, at the cost of some small non-zero blocking probability. In the TSOBS context, the impact of a blocking OTSI will be to reduce the statistical multiplexing performance slightly. Perhaps the most natural choice of delays for a blocking OTSI is the set 1, 2, 4, . . . , $N/2$. This allows any time-slot to be switched to any of the output timeslots, provides small total delay (255 for N=256) and small crossbar size (8×8 for N=256). We have shown that an OTSI with these delays can be

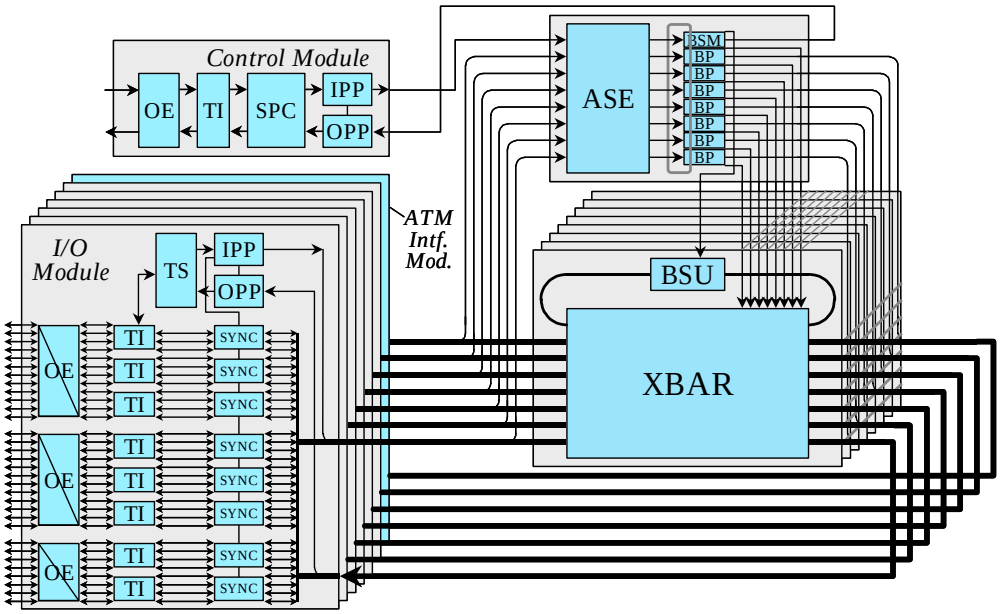

Figure 10. Prototype Burst Switch

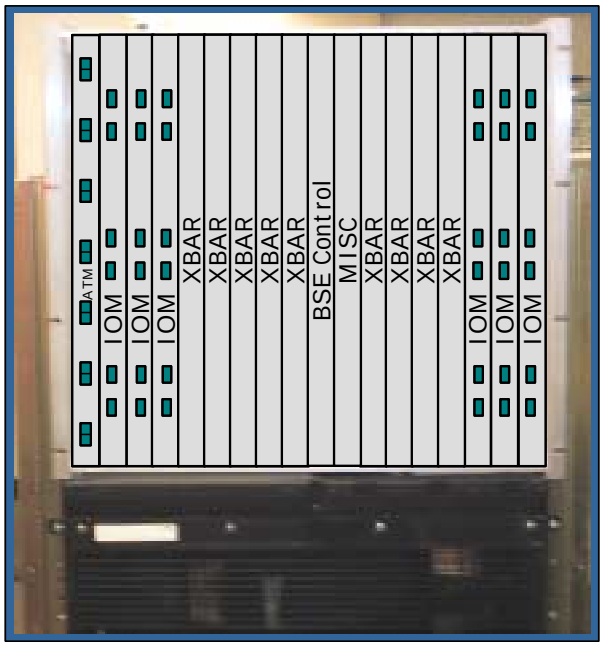

Figure 11. Planned Physical Packaging of Prototype System

operated so as to achieve a small average number of switching operations (<2 under most conditions), and that the impact of blocking on the statistical multiplexing performance is very small.

8. Burst Switch Prototyping Effort

One of the major activities of this project was the development of a prototype burst switch. The purpose of this prototype was to demonstrate the OBS concept and provide an experimental

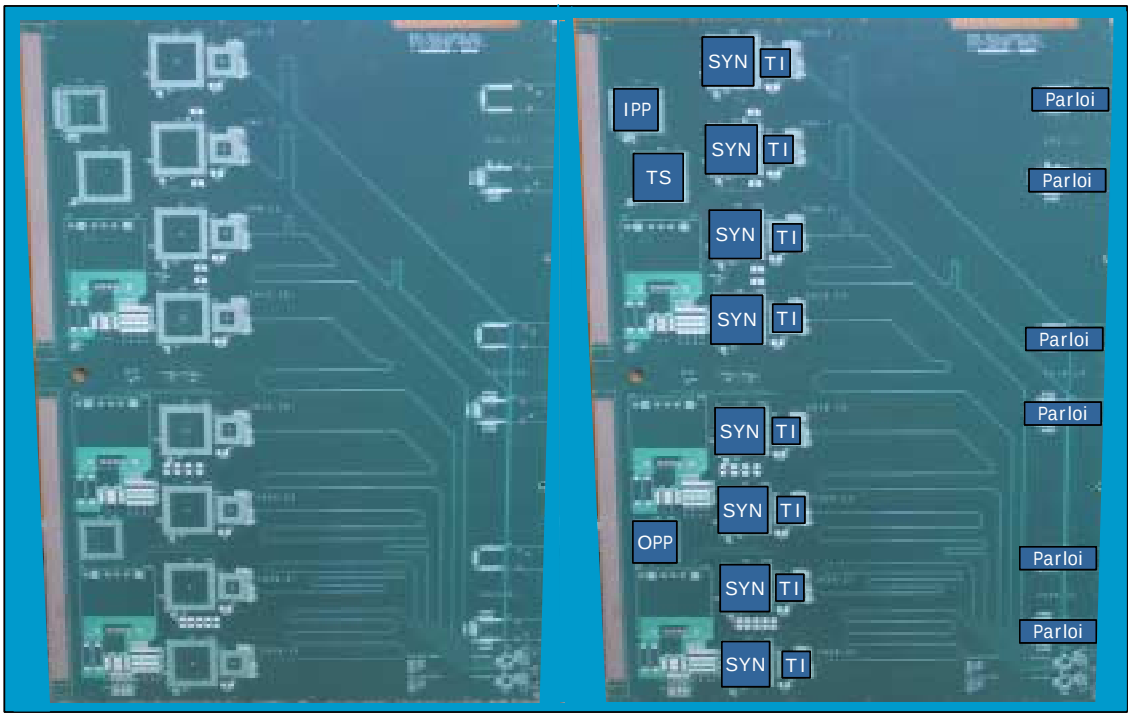

Figure 12. IO Module Printed Circuit Board

vehicle for evaluating different burst scheduling algorithms and burst storage management algorithms. Unfortunately, the prototype was not completed, largely due to a budget cut that affected the entire Next Generation Internet program (NGI) and which took away funds that were needed to complete the fabrication of the prototype and the planned evaluation activity. This section documents the prototyping effort and summarizes what was completed.

Figure 11 shows the structure of the planned prototype system. Since the purpose of the prototype was primarily to demonstrate the control aspects of an OBS router and because the components required for an optical data path would have been prohibitively expensive, both the control and datapath are implemented using electronics. The external links substitute parallel optics for WDM optics. Specifically, each logical OBS link consisted of 32 optical fibers, each operating at 1 Gb/s. Optical ribbon cables are used for sub-groups of up to 12 fibers. The system consists of a total of 18 printed circuit boards. Six of these boards implement IO modules, each terminating one logical OBS link. One board implements an ATM interface terminating 7 OC-48 links and converting streams of ATM cells into OBS data bursts. Ten of the 18 boards implement a seven port Burst Switch Element with an aggregate throughput of well over 200 Gb/s. One of the ten boards implements the control section, while the other nine implement the datapath, which includes a 256×256 crossbar plus the Burst Storage Unit (BSU). The final board, called the Miscellaneous Board (MB) includes a control interface to enable a remote computer to exchange control cells with the prototype system, for the purpose of configuring it and monitoring its performance. The MB also includes timing circuits and circuits to reprogram the many FPGAs on the various circuit boards. The planned physical packaging of the prototype is illustrated in Figure xx. The following sections describe each of the components and document their status at the time the program ended.

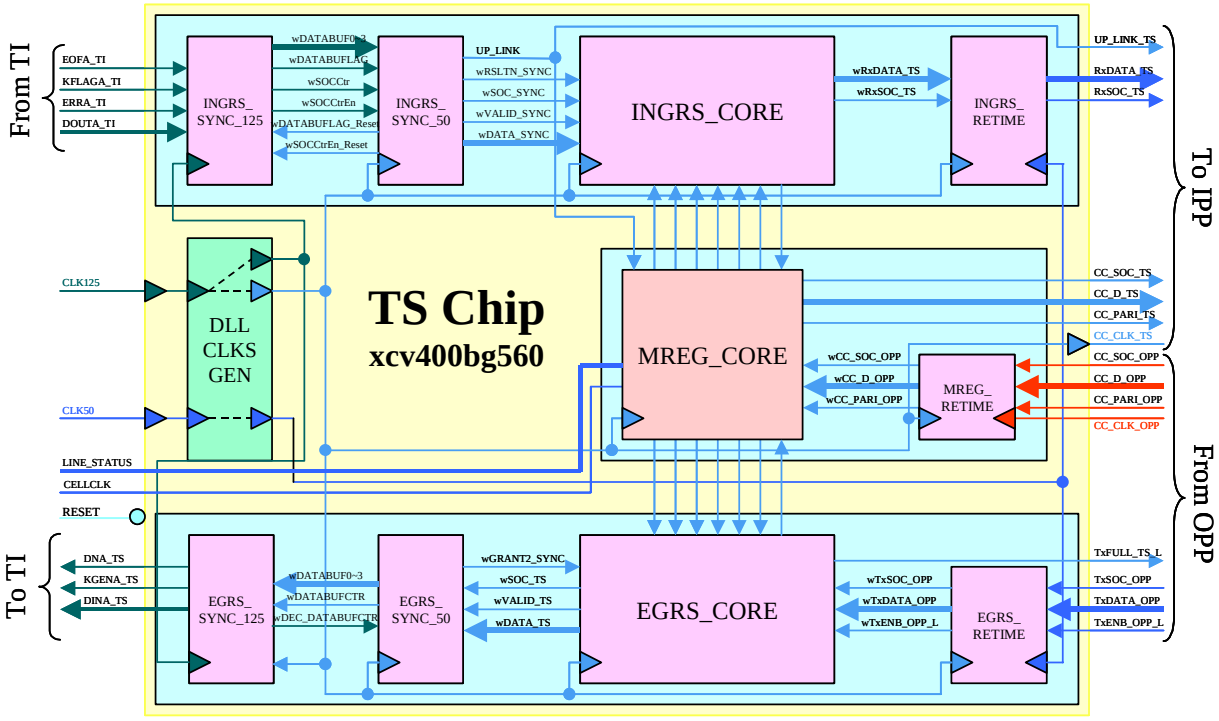

Figure 13. Time Stamp Chip

8.1. IO Module

Each IO module terminates 32 bidirectional optical links operating at 1 Gb/s each. The optical signals are carried on optical ribbon cables with 12 fibers each and terminated using VCSEL-based transmitters and matching receivers. The transmission format uses a standard 8B/10B encoding and is supported by Transmission Interface (TI) components, which terminate four bidirectional channels each. One of the 32 channels carries control information and is separated from the others and connected to the *Time Stamp Chip* (TS) which extracts the BHCs from the incoming data stream, timestamps them with their arrival time on receipt and uses the timestamp to update the offset field on output. The TS chip, forwards arriving cells to an ATM Input Port Processor (IPP), which does a routing lookup before forwarding the BHCs to the control section of the BSE.

The 31 data channels on each logical OBS link are passed through a set of *Synchronization* (SYNC) components which delay the received bursts for a fixed (but programmable) time period to allow for the delays experienced by BHCs in the control subsystem. The SYNC chips then format bursts in a seventeen bit wide data format (the seventeenth bit carries synchronization information) and forwards them to the datapath section of the BSE. On output, bursts from the BSE pass through the SYNC and TI components before being converted to optical form and transmitted on the outgoing link. BHCs are sent from the BSE to an ATM Output Port Processor (OPP), which passes them on to the TS chip and from there to the outgoing control channel of the logical OBS link.

Figure 12 contains a photograph of the printed circuit board for the IO module. The right hand side shows how the major components were to be placed on the board, although the board assembly was never completed. The parallel optical interface (PAROLI) and the transmission interfaces (TI) components were purchased. The ATM OPP was re-used from an earlier project.

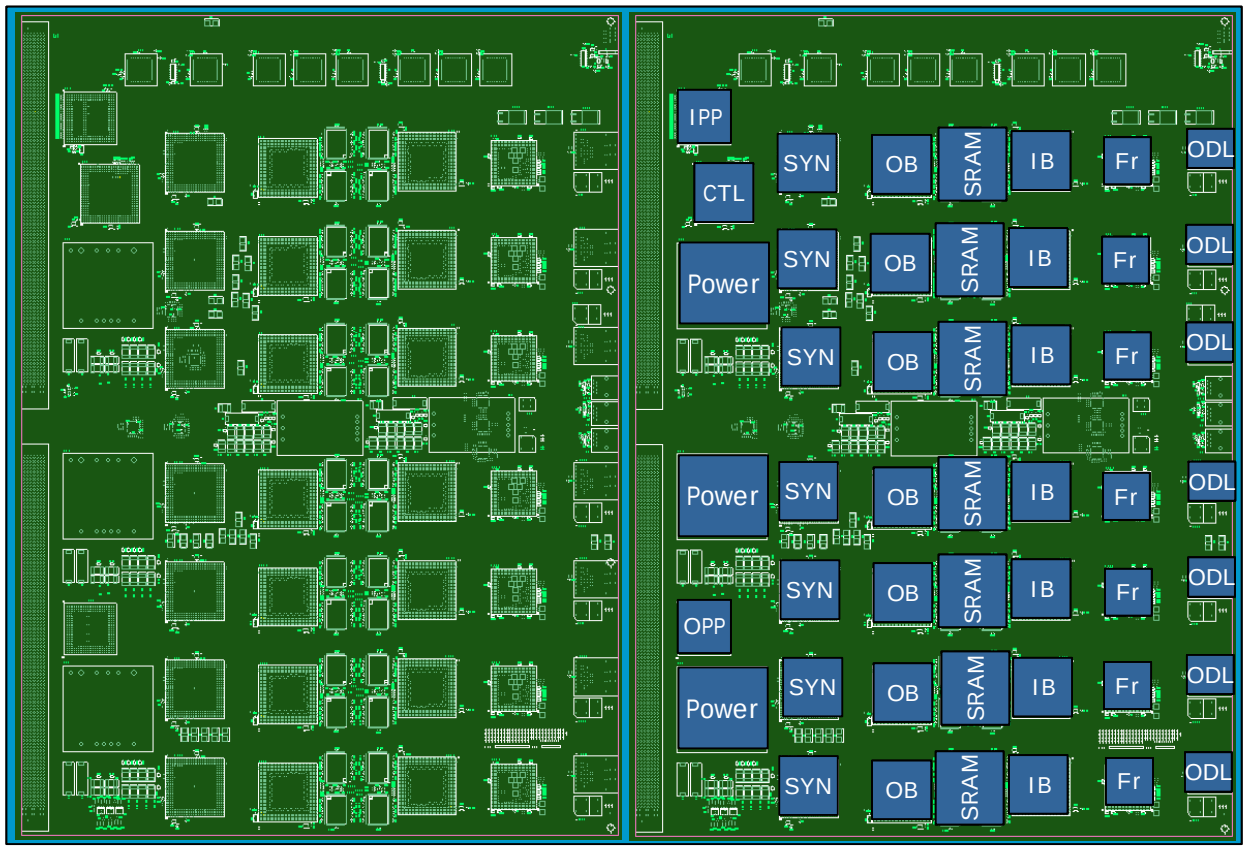

Figure 14. ATM Interface Board.

The IPP is a modified version of a chip originally developed for an earlier project. It is described in the subsequent section. Both the SYNC and TS components were implemented using FPGAs. A block diagram of the TS chip is shown in Figure 13.ATM Interface

The ATM network interface terminates seven OC-48 ATM links and converts streams of ATM cells into OBS data bursts. Specifically, cells belonging to different virtual circuits are buffered in the ATM interface until either a prespecified number of cells has been collected or a prespecified delay has passed since reception of the first cell. At that point, the buffered cells are transmitted as an OBS data burst. On output, received bursts are buffered and forwarded on the appropriate outgoing ATM virtual circuits.

The ATM Interface consists of two major sections, a front end and a back end. The front end comprises seven OC-48 interfaces (ODL), ATM Framers (FR), Input Buffers (IB) and Output Buffers (OB), each with an associated set of SRAM components for storing bursts. The back end comprises a set SYNC components (identical to those in the IO modules), an ATM IPP and OPP and a Controller, which connects to all the IB and OB components, as well as to the IPP and OPP. The IB, OB and Controller were all implemented using FPGAs. Figure 14 shows the layout of the printed circuit board for the ATM interface. The placement of the major components on the board is shown in the right-hand view.

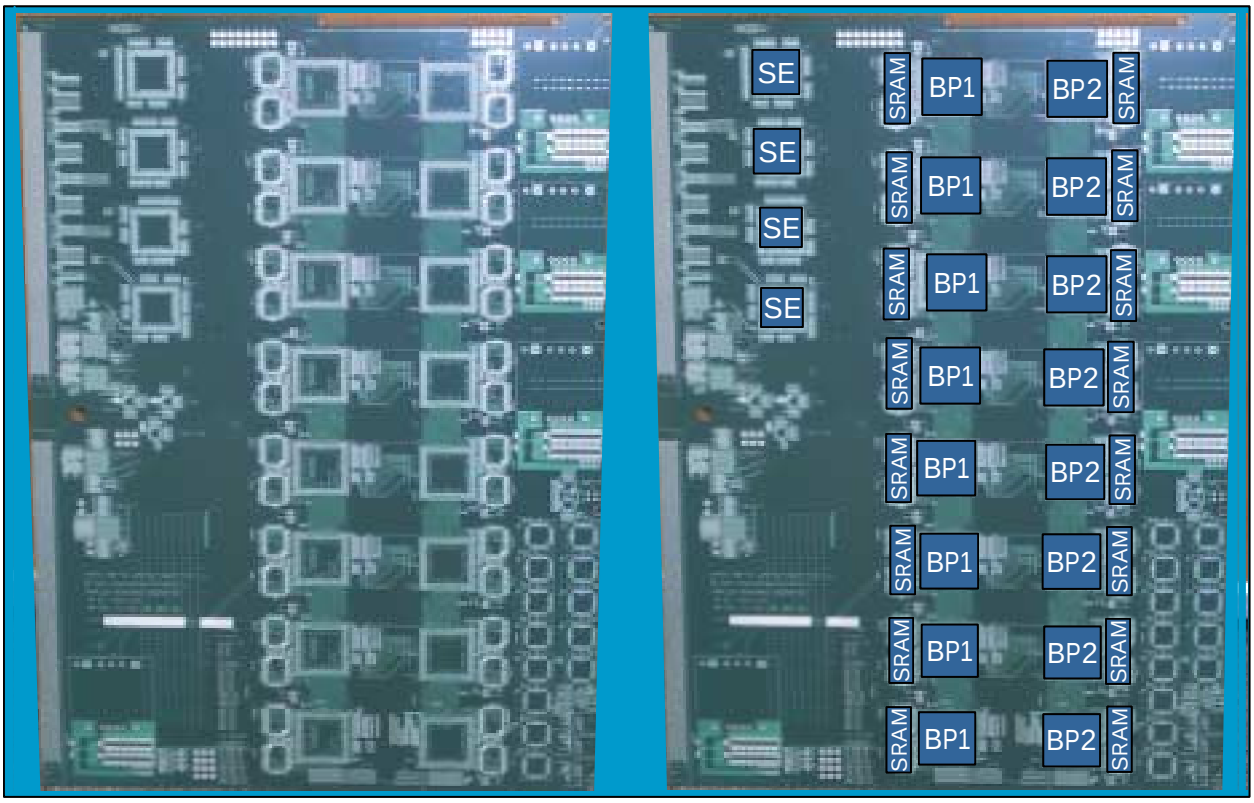

Figure 15. BSE Control Board

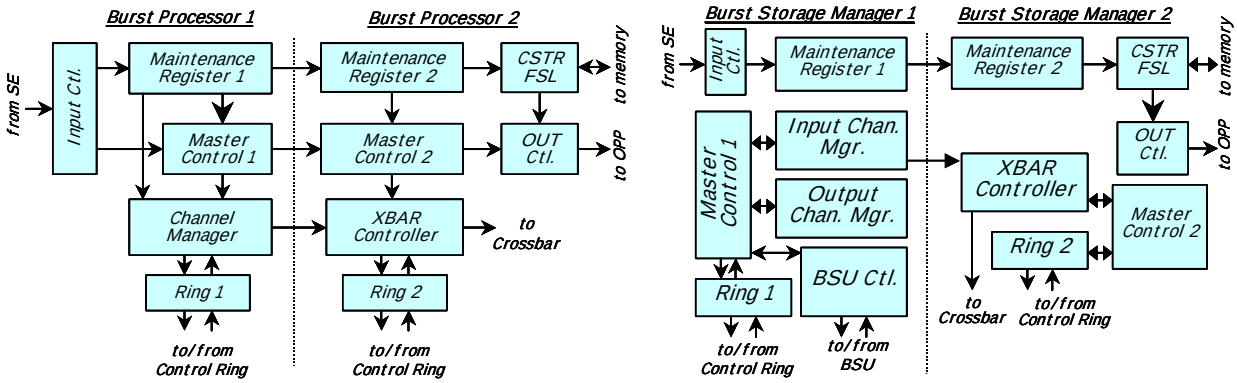

Figure 16. Burst Processor and Burst Storage Manager

8.2. Burst Switch Element Control

The BSE Control board contains an eight port ATM Switch Element (implemented with four custom ASICs), seven Burst Processors (implemented with two FPGAs, each) and a Burst Storage Manager (also implemented with two FPGAs). The BPs schedule the transmission of bursts on the outgoing links and the BSM manages the use of the shared burst storage. Figure 15 contains a photograph of the BSE Control Board. The right hand side of the figure also shows the planned placement of the major components.

Figure 16 shows block diagrams of the Burst Processor and Burst Storage Manager. Arriving cells pass through the Input Controller and go to the Maintenance Register Blocks to the Cell

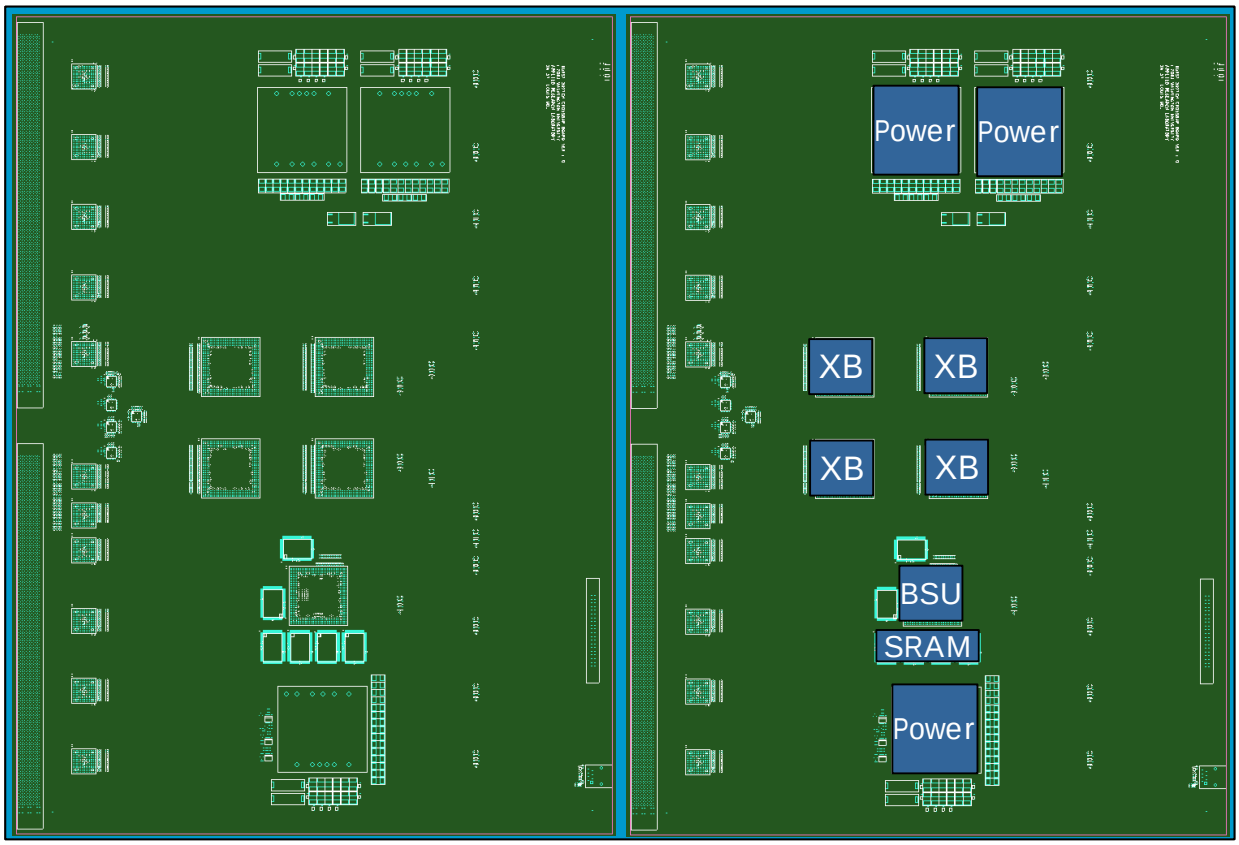

Figure 17. Crossbar Board

Store in the off-chip memory. The essential control information from the BHC is passed to the Channel Manager, which implements a horizon scheduler. When the channel manager makes a scheduling decision, it passes commands to the crossbar controller, which include the time at which the requested operation is to be performed. The control ring is used by the BPs to communicate with the BSM. The BSM has a similar overall structure. The biggest difference is that it has two Channel Managers, one for the ingoing and one for the outgoing channel.

8.3. Crossbar

The Crossbar Board is shown in Figure 17. Each board contains a two bit slice of the crossbar. Two FPGAs on each board implement a 256×256 port crossbar, each chip implementing a 256×128 section of one crossbar. The crossbar has a pipelined data path and a unique control structure that allows subgroups of 32 outputs to be controlled independently of one another. The eight separate control interfaces are used by the BPs and BSM on the control board so that each can control access to the crossbar outputs corresponding to the link it is responsible for.

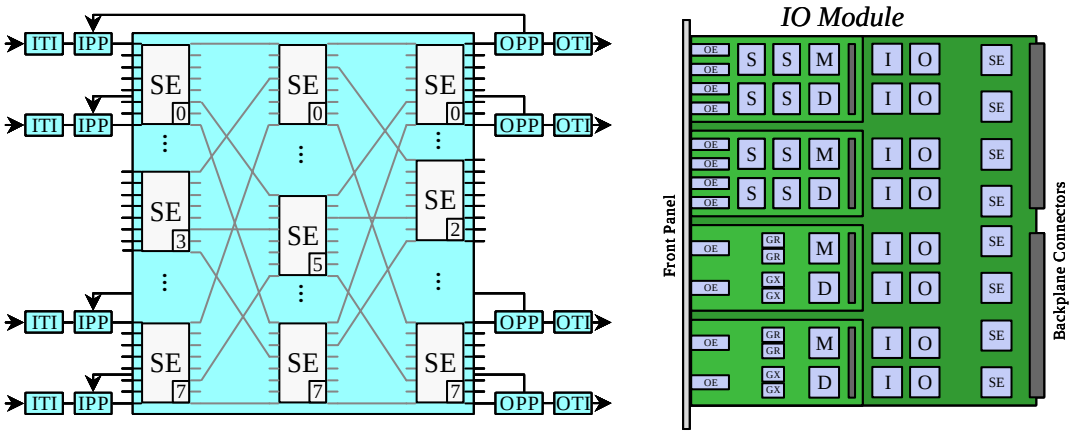

Figure 18. 160 Gb/s ATM Switch

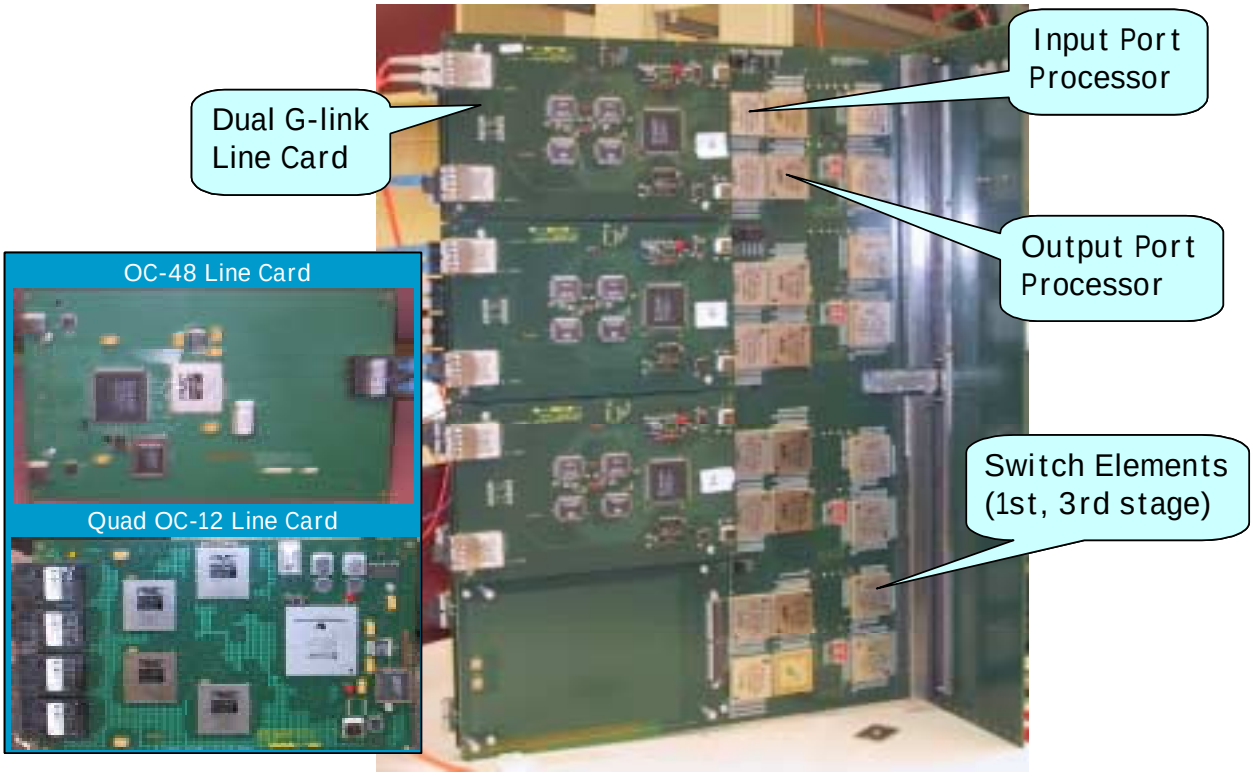

Figure 19. ATM Switch IO Module in Test Fixture

9. 160 Gb/s ATM Switch

While the primary focus of this project was Burst Switching, the project also included the design and implementation of a 160 Gb/s ATM Switch, illustrated in Figure 18. This system was designed with 64 ports, each capable of hosting a line card with an IO bandwidth of up to 2.4 Gb/s. Three different line cards were implemented, an OC-48 Line Card, a Quad OC-12 Line Card and a Dual Gigabit Line Card. The system is built around a three stage interconnection network, with shared buffer, eight port Switch Elements. Three custom ASICs are used in the

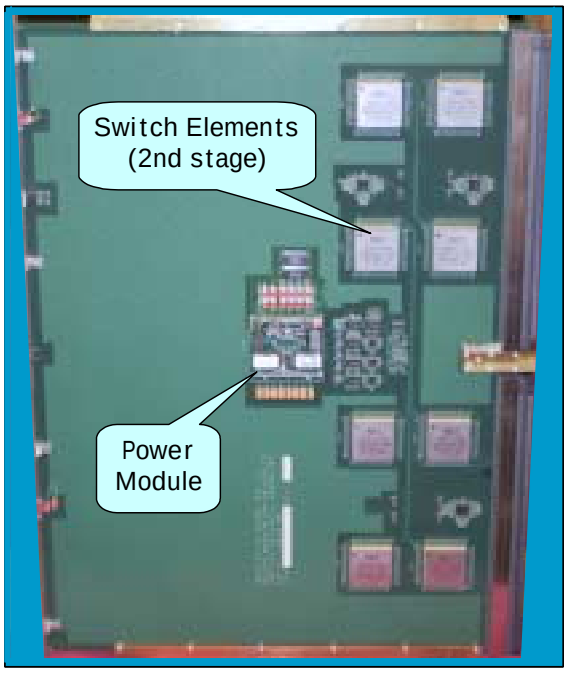

Figure 20. Switch Element Board

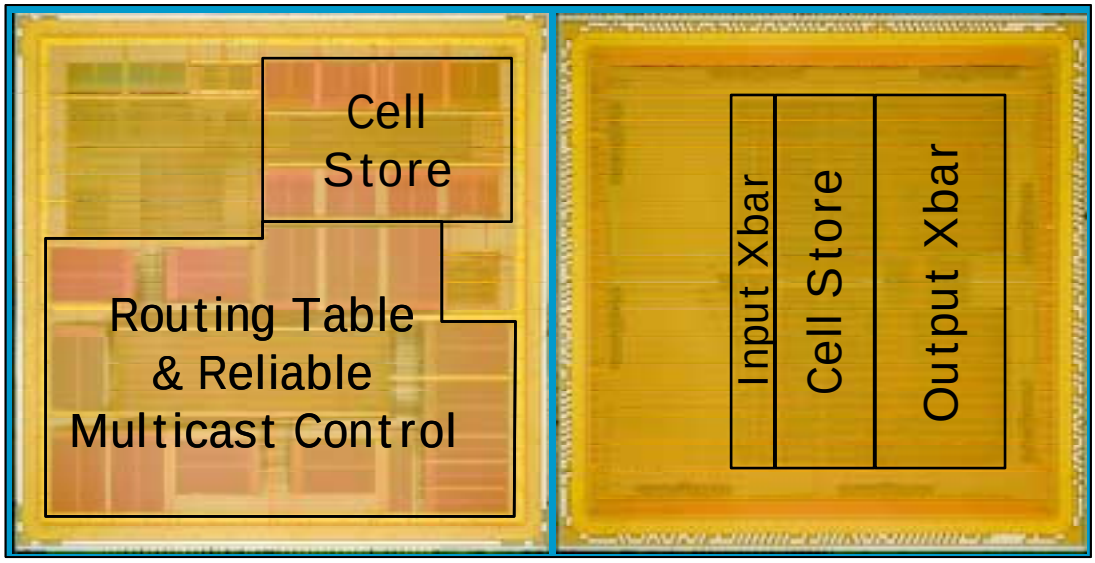

Figure 21. Input Port Processor (left) and Switch Element ASICs

design. Two of these were modified versions of designs developed in an earlier project. The third (the Output Port Processor) was simply adopted directly from the earlier project.

The overall system comprises twelve circuit boards. Eight of these implement IO Modules, each of which hosts eight Input Port Processors (IPP) and Output Port Processors (OPP) and their associated line cards. The IPPs perform ATM virtual circuit lookups on received cells. The OPPs buffer outgoing cells waiting to be sent to an output link. The IO modules also include two eight

port Switch Elements (SEs). One of these is a first stage SE and the other is a third stage SE. The second stage SEs are carried on a set of four SE circuit boards.

Figure 19 shows the IO module in a test fixture, and highlights the various components used to implement it. Figure 20 shows the Switch Element Board and Figure 21 shows the Input Port Processor and Output Port Processor ASICs.

10. Summary

Further details concerning the project can be found in the progress reports and publications included in the reference list below. The Terabit Burst Switching Project has made important contributions to our understanding of the design of practical OBS systems. It has resulted in a number of important design innovations and has provided new insights into the issues and tradeoffs that impact the cost and performance of these systems. At the present time, the high cost of wavelength conversion constitutes a significant obstacle to the development of cost-effective OBS routers. Continuing progress will require either dramatic changes in the cost of wavelength conversion relative to electronic switching components, or the development of architectural variants like TSOBS that do not require wavelength conversion.

REFERENCES

[AN83] Amstutz, Stanford. "Burst Switching --- An Introduction," *IEEE Communications*, 11/83.

[BA97] Barry, Richard A. "The ATT/DEC/MIT All-Optical Network Architecture," In *Photonic Networks*, Giancarlo Prati (editor), Springer Verlag 1997.

[CH97] Chaney, Tom, J. Andrew Fingerhut, Margaret Flucke and Jonathan Turner. "Design of a Gigabit ATM Switch," *Proceedings of Infocom*, 4/97.

[CH00] Chen, Yuhua and Jonathan Turner. "WDM Burst Switching for Petabit Capacity Routers," *Proceedings of Milcom*, 1999.

[CH03] Chen, Yuhua. "Design of Optical Burst Switching Systems," Doctoral dissertation, Washington University Electrical Engineering Department, expected 5/2003.

[CO90] Cormen, Thomas, Charles Leiserson, Ron Rivest. *Introduction to Algorithms*, MIT Press, 1990.

[CO02] Coffman, K. G. and A. M. Odlyzko. "The Size and Growth Rate of the Internet," In *Optical Fiber Telecommunications IV B: Systems and Impairments*, I. P. Kaminow and T. Li, eds. Academic Press, 2002.

[GA97] Gambini, Piero. "State of the Art of Photonic Packet Switched Networks," In *Photonic Networks*, Giancarlo Prati (editor), Springer Verlag 1997.

[GU96] Gustavsson, Mats. "Technologies and Application for Space Switching in Multi-Wavelength Networks," In *Photonic Networks*, Giancarlo Prati (editor), Springer Verlag 1997.

[IK96] Ikegami, Tetsuhiko. "WDM Devices, State of the Art," In *Photonic Networks*, Giancarlo Prati (editor), Springer Verlag 1997.

[MA96] Masetti, Francesco. "System Functionalities and Architectures in Photonic Packet Switching" In {*Photonic Networks*, Giancarlo Prati (editor), Springer Verlag 1997.

[PR95] Prucnal, Paul and I. Glesk. "250 Gb/s self-clocked Optical TDM with a Polarization Multiplexed Clock," *Fiber and Integrated Optics*, 1995.

[RA02] Ramamirtham, Jeyashankher and Jonathan Turner. "Design of Wavelength Converting Switches for Optical Burst Switching," *Proceedings of Infocom 2002*, 6/02.

[RA03] Ramamirtham, Jeyashankher and Jonathan Turner. "Time SlicedOptical Burst Switching," to appear in *Proceedings of Infocom 2003*, 7/01.

[ST96] Stubkjaer, K. E., et. al. "Wavelength Conversion Technology," In *Photonic Networks*, Giancarlo Prati (editor), Springer Verlag 1997.

[TU94] Turner, Jonathan S., "An Optimal Nonblocking Multicast Virtual Circuit Switch," *Proceedings of Infocom*, June 1994.

[TU96] Turner, Jonathan S. and the ARL and ANG staff. "A Gigabit Local ATM Testbed for Multimedia Applications," Washington University Applied Research Lab ARL-WN-94-11.

[TU98a] Turner, Jonathan S. "Terabit Burst Switching," Washington University Technical Report, WUCS-98-17, 1998.

[TU98b] Turner, Jonathan S. "Terabit Burst Switching Progress Report (12/97-3/98)," Washington University Technical Report, WUCS-98-16, 1998.

[TU98c] Turner, Jonathan S. "Terabit Burst Switching Progress Report (3/98-6/98)" Washington University Technical Report, WUCS-98-22, 1998.

[TU98d] Turner, Jonathan S. "Terabit Burst Switching Progress Report (6/98-9/98)" Washington University Technical Report, WUCS-98-30, 1998.

[TU98e] Turner, Jonathan S. "Terabit Burst Switching Progress Report (9/98-12/98)" Washington University Technical Report, WUCS-98-31, 1998.

[TU99a] Turner, Jonathan S. "Terabit Burst Switching," *Journal of High Speed Networks*, vol. 8, no. 1, 1999.

[TU99b] Turner, Jonathan S. "WDM Burst Switching," *Proceedings of INET*, San Jose, CA, 6/99.

[TU99c] Turner, Jonathan S. "WDM Burst Switching for Petabit Capacity Routers," *Proceedings of Milcom*, Atlantic City, NJ, 11/99.

[TU99d] Turner, Jonathan S. "Terabit Burst Switching Progress Report (1/99-6/99)" Washington University Technical Report, WUCS-99-21, 1999.

[TU99e] Turner, Jonathan S. "Terabit Burst Switching Progress Report (7/99-12/99)" Washington University Technical Report, WUCS-99-32, 1/2000.

[TU00a] Turner, Jonathan S. "WDM Burst Switching for Petabit Data Networks" *Proceedings of the Optical Fiber Conference*, 3/2000.

[TU00b] Turner, Jonathan S. "Terabit Burst Switching Progress Report (1/00-6/00)" Washington University Technical Report, WUCS-00-18, 7/2000.

[TU00c] Turner, Jonathan S. "Terabit Burst Switching Progress Report (7/00-9/00)" Washington University Technical Report, WUCS-00-28, 10/2000.

[TU01a] Turner, Jonathan S. "Terabit Burst Switching Progress Report (10/00-3/01)" Washington University Technical Report, WUCS-01-09, 5/2001.

[TU01b] Turner, Jonathan S. "Terabit Burst Switching Progress Report (4/01-6/01)" Washington University Technical Report, WUCS-01-23, 7/2001.

[TU02] Turner, Jonathan S. "Terabit Burst Switching Progress Report (4/01-6/01)" Washington University Technical Report, WUCS-02-12, 5/2002.